

SHAPES — IN — COLOR

HARRY MONEYHUN

Create your own animated shapes and figures . . . paint a landscape . . . design an alphabet! Explore the limits of your imagination with this precision shape plotting tool. Command the power of Hi-Res graphics to produce animations, games, graphic presentations, and more.

With *Shapes in Color*, you develop shape tables quickly and accurately. Shapes are formed on an enlarged plotting grid so you can easily visualize your creation. Graphic backgrounds can be "painted" with free-hand brushstrokes, and then used with moving shapes to generate striking effects. Completed backgrounds and shapes are saved on disk to be reloaded for use in your programs.

Detailed documentation describes techniques for writing original programs with the shapes and backgrounds you have created, making mastery of animation and High-Resolution graphics simple and fun for the beginner.

SHAPES — IN — COLOR



*The Creative Tool That Lets You Design
High-Resolution Animation and Graphics*

HAYDEN SOFTWARE

SHAPES IN COLOR

Harry Moneyhun

HAYDEN SOFTWARE

600 Suffolk Street, Lowell, Mass. 01853

Copyright (c) 1982 by Hayden Book Company, Inc.

Apple Computer, Inc., makes no warranties, either express or implied, regarding the enclosed computer software package, its merchantability, or its fitness for any particular purpose.

Limited Warranty Subject to the below stated limitations, Hayden Software Company hereby warrants that the program(s) contained in this unit will load and run on the standard manufacturer's configuration of the computer listed for a period of ninety (90) days from the date of purchase. Except for such warranty this product is supplied on an "as is" basis without warranty as to merchantability or its fitness for any particular purpose. The limits of warranty extend only to the original purchaser.

Neither Hayden Software Company, not the author(s) of this program are liable or responsible to the purchaser and/or user for loss or damages caused, or alleged to be caused, directly or indirectly by this software and its attendant documentation, including (but not limited to) interruption of service, loss of business or anticipatory profits.

To obtain the warranty offered, the enclosed purchaser registration card must be completed and returned to Hayden within ten days of purchase.

This is a fully copyrighted work and as such is protected under the copyright laws of the United States of America. According to these laws, consumers of copywritten material may make copies for their personal use only. Duplication for any other purpose whatsoever would constitute infringement of copyright laws and the offender would be liable to civil damages of up to \$50,000 in addition to actual damages, plus criminal penalties of up to one year imprisonment and/or a \$10,000 fine.

Contents

Introduction: Shapes in Color	1
Part One: Creating Hi-RES Graphics	
Shapes	7
Background.....	15
Animation using Shapes and Backgrounds	20
Color and High Resolution Graphics	27
Part Two: Shape File Utilities	
Modify	35
Convert	41
Hardcopy	42
Part Three	
Appendix A: Command Reference.....	44
Appendix B: The Candle Shape Project.....	50

SHAPES IN COLOR

SHAPES IN COLOR is an easy-to-use, precision shape-drawing tool. Amateurs and professionals can create imaginative graphics, from animation to unique typography, in living, high resolution color! A variety of color shapes and character fonts can be constructed and saved in a shape table for use in other programs. Create imposing beasts and monsters for adventure games; design original shapes and invent characters for animation; even construct custom type-faces. This system lets anyone use the Apple's high resolution (HI-RES) graphics capabilities with ease.

While many technical details relating to the use of graphics are explained in this manual, some familiarity with the Apple is assumed. The user should be able to program in BASIC. For further explanation of programming terms not explained here, refer to the Apple reference manuals.

The following equipment is needed to use SHAPES IN COLOR:

- An Apple II Plus with Applesoft;
- 48k of memory;
- One Disk Drive;
- A color television monitor.

This manual has three parts. Section One is called *Creating Hi-Res Graphics* and describes programs for creating shapes and graphics. Section One also contains two chapters on how to use the shapes created, with an emphasis on animation techniques. While the instructions for using the SHAPES and BACKGROUND programs comes first, it is likely be more helpful to review the sections on *Animation* and *Color and High Resolution Graphics* before actually designing any shapes or backgrounds.

Section Two, *Shape File Utilities*, explains the programs which modify previously created shapes. This part also includes more information on the structure of the color shape tables used by these programs.

Finally, Section Three contains the two appendices. The first is a quick-reference guide to both HI-RES and SHAPES IN COLOR commands; the second, a step-by-step example of how to plot a shape using SHAPES IN COLOR.

The SHAPES IN COLOR disk contains six programs:

The SHAPES program, with its unique plotting grid and color guide markers, lets you construct shapes and designs limited only by imagination and the capabilities of the machine.

The BACKGROUND program is for designing backgrounds for games, animations, or full-screen title pages.

The program MODIFY is used to transfer shape tables from one disk to another and for constructing a table of shapes by combining shapes from one or more existing tables.

CONVERT will convert a standard shape, as described in the Applesoft BASIC manual, into a 'color' table in the format used by SHAPES IN COLOR. This special format is one of the things about SHAPES IN COLOR that makes it easier for the programmer.

The HARDCOPY programs lets you list on a printer the decimal value of each byte in a shape table.

The DEMO illustrates some of the possibilities that can be realized using the SHAPES IN COLOR programs described above.

To use the system, simply insert the disk in the drive and power up

the system. The disk will "boot," and a title page with an option menu is displayed. Select the program to be run by pressing the corresponding number key. (Do not hit **RETURN**.)

Some important facts to remember when using SHAPES IN COLOR:

- 1 All commands entered by the user are displayed in **bold face** in this manual.
- 2 Some commands used in these program are executed immediately upon typing a single keystroke; others require that **RETURN** be pressed before any action is taken by the computer. The manual will note which of these two types a given command is. Typing an extraneous **RETURN** sometimes causes unexpected results.
- 3 SHAPES IN COLOR is not a menu-driven system. That is, when one program ends, control does not automatically return to the main option table displayed when you first "boot" the system. When one program has ended, it is of course possible to run another program without re-booting the disk. Type **RUN program name** , and the desired program is run. (The exact **program names** to use are the ones given in the descriptions above.) Or, type **RUN SHAPES IN COLOR** to return to the main menu.
- 4 All file names must begin with a letter, and may contain up to 30 alphanumeric characters.
- 5 Most of the programs on this disk either load or save shapes tables from a disk. To store the shapes created by these programs, a separate initialized 'work disk' should be used. The SHAPES IN COLOR program disk should not be used for storage of user-created shapes.

Part One
Creating Hi-Res Graphics

SHAPES

Beginning a Shape Table

A shape table consists of a number of shapes defined by a sequence of plotting vectors. The SHAPES program lets you create your own shapes and compile them into a table. Later, these shapes can be used in an Applesoft BASIC program, and drawn anywhere on the HI-RES screen, in different colors, sizes, and angles.

SHAPES may be chosen from the main menu by pressing **1** (do not press **RETURN**). It can be run from BASIC by typing **RUN SHAPES IN COLOR** at the keyboard. After the program is loaded, the program disk should be removed, and a "work disk" should be inserted.

Once the program begins running, the first prompt is for the name of the shape table to be either created, added to, or reviewed.

By pressing **RETURN** in response to this prompt, the disk is cataloged. (That is, a directory of all the file names on the disk is displayed.) The prompt for the shape table name then reappears.

A **CTRL-Q** and **RETURN** at this point exits the program.

If a valid file name is entered, the program looks for that file on the disk. If a binary file with that name is found, it is loaded into memory as a shape table. If the loaded file is not in the format of a color shape table, the message NOT A COLOR TABLE is displayed. (Refer to the section on converting shape tables.)

If the table loaded is full, that is, there is a shape for every space allocated for one in the shape table directory, then the user is presented with the option of reviewing the table. A response of **Y** causes the shapes to be displayed in groups of twenty. (Larger shapes may overlap.) An **N** response at this point exits the program. If the table is not full, then the "plotting grid" used to create shapes is displayed, and the program moves on to the shape-creating options.

If the shape table file is not found on the disk, the program starts a new shape table and prompts for the number of shapes to be to created.

The number of shapes selected must be greater than one, and less than 200. Pressing **RETURN** accepts the default of ten shapes. Once the number of shapes has been selected, the program can set up its shape table directory and proceed to the plotting grid.

Plotting Grid

A violet plotting grid, surrounded by a violet border, is drawn on the screen. The size of the grid defaults to 15 x 15 unless changed by the **NEW GRID** option (**N**), after which it defaults to the last size used.

Directly above the green field grid in the lower right corner, a small rectangle is displayed to illustrate the actual size of a shape that would fill the entire plotting grid. The plotting grid is sectioned by dashed lines in 5 x 5 squares to simulate standard graph paper. The center of each square is between spaces if the width is an even number, or on a space if it is an odd number. Keep in mind that the plotting grid, with its markings, is strictly a guide to shape building and has nothing to do with the shape itself or its size.

In the lower right corner of the screen is a green "field grid." Each time the cursor is moved out of the display window (or field), it

"wraps-around" to the other border of the display. The marker in the field grid indicates the new field being plotted in. If the field marker leaves the field grid, it disappears until the plotting marker has returned to a field that is within range of the field grid display. (This has no effect on the shape being drawn.) When the cursor is moved back to the original field, the marker in the field grid is returned to the center.

Color Guides

Above the plotting grid is a color chart that shows the colors that may be plotted in the odd and even vertical columns. If a color other than white has been chosen, only that color appears in the chart. If either of the two whites are selected as the shape color, then a plot in a given column produces the color that is at the top of that column. If, however, plots are made in adjacent columns, that is, when two colors are plotted side by side, white is obtained.

As you can see, due to the way in which the Apple generates color, confusing results are possible when plotting shapes and lines in color. The color guides in the SHAPES program are designed to make it easier for the programmer to predict the effects that will result when used in a program.

In the upper right corner is a 'color bar' that indicates the shape color that has been selected. If it is one of the two whites, then the two component colors are also displayed. (The 'component' colors being the same that appear on the top of each grid column.)

The color bar indicates the initial "color value" in the shape table. Once a shape has been created and is used in a program, this 'initial color' can be retrieved from the color directory and used to provide the proper value for the **HCOLOR** statement used when drawing a shape table. How to change this initial color is explained below.

Previous Shapes

If there is a shape already in memory, it is displayed beneath the color bar. If more than one is in memory, the first and last are displayed, the last one in the table beneath the first. If either of these shapes are large, they may extend into the plotting grid. If this is objectionable, type **T** (for toggle) and **RETURN**. The grids

are redrawn, without the first and last shape; if **T** is keyed again, they will be re-displayed.

Set-up options

The set-up options allow selection of various starting values assumed by the program, such as initial color.

A numerical response to the **X?** prompt indicates the choice for the x-coordinate of the starting position. Because the color of a shape when drawn on the HI-RES screen depends on the x-coordinate, the program will only accept an odd value. Neither the x nor the y will be accepted if it is out of the display area. The default value for x is 1 (the leftmost grid column) until changed, then it is the last x. The y default value is the bottom line of the plotting grid, unless it is changed, then it is the last y. A shape may start anywhere inside the display window. The actual range of values accepted outside the grid depends on the size of the grid.

Grid Size

Typing **N** instead of a number allows re-specification of the dimensions of the plotting grid. The HI-RES display is cleared and the size of the plotting grid may be selected. A **RETURN** accepts the default values. For x the default value is 15; the default value for y is the value of x, but not larger than 25. An x value from 5 through 40 is allowed. If a value less than 5 is entered, the grid defaults to 5 x 7, the standard character font size. If a value larger than 40 is chosen for x, the grid will default to the maximum size of 40 x 25.

Initial Color

The initial color is changed by typing **G, V, W, O, B, or A**, representing Green, Violet, White, Orange, Blue, and Alternate white. The shape color may be chosen when the **X?** prompt is asking for the starting coordinates by keying one of the above letters. In most cases, either **W** (white) or **A** (alternate white) will be used. This allows creation of a shape in any or all of the three colors indicated by the color bar. The color selection is not an absolute limitation but rather a convenience in plotting and programming. Choosing an initial color places the corresponding color-number in the 'shape color directory,' where it may be

retrieved from within a program. By drawing the shape only in odd-numbered columns, the shape is displayed by the Apple in the colors originally intended by the programmer. If desired, however, the shape may be drawn in another color simply by specifying a new **HCOLOR** within the program the shape is used in.

Review

To review the previously-created shapes in the shape table currently in memory, type **R**, which causes them to be displayed in groups of twenty. Again, larger ones may overlap.

Quit

Typing **CTRL-Q** exits the program, with the variables set for proper re-entry. The program may be restarted with the variables intact by typing **GOTO 99**.

Color Plotting

Once the shape color, grid size, and starting coordinates have been set up, and the plotting mode has been entered, shape construction begins. If the default values have been used, the shape color is white, (which included green and violet), the grid size is 15 x 15, and the starting position is in the lower left of the plotting grid.

As the plotting is being done, the actual-sized shape is drawn below the grid and also on HI-RES page two, which normally is not displayed.

The following commands are available in the plotting mode:

J - LEFT

Moves cursor LEFT

K - RIGHT

Moves cursor RIGHT

I - UP

Moves cursor UP

M - DOWN

Moves cursor DOWN

P - PLOT

Plots at cursor position

Z - View HI-RES Page Two

The plotting grid is displayed on HI-RES page one. While the moves and plots are being made the actual-sized shape will be drawn in HI-RES at the bottom of the page; however, the plotted shape will not extend beyond the space allotted for it. For this reason the shape will also be plotted on HI-RES page two at the same time. Touching the **Z** key will switch to page two showing the entire shape as it appears so far. Plotting may continue on page two. To find out where the cursor is after making some moves on page two, touch the **SPACE** bar and a small cross hair appears momentarily to locate the cursor position. The color of the vertical hair indicates the color of the column.

RETURN - View HI-RES Page One

While on page two, the **RETURN** key will flip the display back to page one, with the plotting grid intact and the moves and plots on the grid as they would have been if they had been entered while on that page.

? - Length of Table

The length of the binary table may be checked by touching **?**. This length is continually being changed by moves and plots. The operation has no effect on the shape.

C - Cancel Shape

To cancel the shape and redraw the grid, press **C**. This restarts the shape with the selection of the x and y starting coordinates. Hitting **RETURN** twice will accept as the default values those just used.

E - Erase Last Plot

While in the plotting mode, if no move has been made after the last plot, the **E** key will erase the last plot.

! - Clear Plotting Grid

If the plotting grid becomes cluttered with plots from the 'wrap-

around,' touch the **!** key, and the entire window framing the plotting grid will be cleared and re-drawn without disturbing the rest of the display and without effecting the shape being created. To clear the shape being built, use the **C** option.

Plot Marker Colors

The plot marker is a white block (number 3 to the computer). If the shape is so large that the cursor has gone into the next field, the plot marker, due to the wrap-around feature, may be confused with the old markers. A different color may be chosen for the marker. Type the number of the color to change it: **1=Green, 2=Violet, 3=White, 5=Orange, 6=Blue, and 7=Alternate White.** The color of the marker has nothing to do with the shape being drawn.

F - Finish

Pressing **F** will exit the plotting mode. The program then prompts to save the shape table to disk. Be sure a 'work disk' for storing the shape is in the drive before selecting this option. A **Y** in response to this prompt will save the shape table named to the disk; the program then returns to plot the next shape if there are any left in the directory. Any other response (including **RETURN**) will not save to disk, but instead proceeds to the next shape to be created.

About Scale

If a scale factor larger than one is used in a program with a shape from a shape table, it is important to know something about how scaling works. If **SCALE=4** is used in a program, the shape created will not appear just as it was drawn, but will be 4 times the size. Further, the plotted dots are not increased in size, but are converted into a line, and the line extends the direction of a move after a plot. Suppose the following keys were used to draw a 'J': **I-P-K-I-P-K-I-P-K-I-P-K-I-P-K-I-P**. With **SCALE=1** the shape would appear as a slant. However, with **SCALE=4** the shape will be a series of horizontal lines like the treads on a staircase. Now suppose **I-P-I-K-P-I-K-P-I-K-P-I-K-P-I-K-P** had been used. With **SCALE=1**, again there is the same slant as before. But with **SCALE=4**, drawing this shape will produce a series of vertical lines, like the risers of a staircase. Using the following keys: **I-P-K-P-I-P-K-P-I-P-K-P-I-P-K-P-I-P-K-P-I-P-K-P**, with **SCALE=1**, the result is a 'heavier'

slant. With **SCALE=4** the shape appears as a staircase with both risers and treads, and the last line will be horizontal because the last plot (**P**) was followed by a horizontal move (**K**).

To see the effect of scaling, **RUN** the program on the disk named **Z**. The letter **Z** from the table **ABC3** is drawn in three colors, white, green and violet; and in two scales, 1 and 4. When **ABC3** was constructed, a move was made after each plot. Before using **SCALE** in a program, it is wise to do some experimenting.

When creating a shape, keep in mind it may be desirable for the shape to end at a specific point. To do this, move, with or without plotting, back to the starting position or to the position where the next shape will be drawn. If the shape ends at the starting position, then when the shape is drawn or xdrawn in a program, the next **DRAW** will not need parameters. The shape will **DRAW** or **XDRAW** at the place where the last shape ended.

BACKGROUND

Creating Backgrounds

The **BACKGROUND** program was designed to set up a static background for use with the movable shapes created by the **SHAPES** program. It can also be useful for such applications as the construction of title pages and other graphics displays. It should be noted that a HI-RES binary file takes up 34 sectors on the disk, which is a lot. If a title page consists of a few characters and very little design, it may be better to prepare a shape table and write a short program to display the shapes if disk space is a consideration.

Once a HI-RES page has been set up, either by designing a background, or by displaying shapes, the entire HI-RES page can be saved onto the disk in either direct (keyboard) or indirect (program) modes with the instruction: **BSAVE filename, A8192, L8192**. The **BACKGROUND** program does this automatically if the **F** option is keyed.

Using the BACKGROUND Program

When the second option is chosen after booting the disk, or, if **BACKGROUND** is run from the keyboard, a title page is displayed

along with a reminder that typing an asterisk (*****) will display a table of command options. The program then asks for the name of a background file. If a legal file name is entered, the program looks for the file, and if found, loads it and displays the image on the page one HI-RES screen (full screen). The cross hair cursor will momentarily appear centered near the bottom.

If the file is not found on the disk, the program assumes that a new background image is to be built, clears the HI-RES page to black, and momentarily displays the cursor position.

If an illegal file name is entered (for example, one that doesn't begin with a letter), the disk will be cataloged.

If **RETURN** alone is keyed, the program will work with the page one image that is in memory, and display it. If there actually is no HI-RES image in memory, the screen appears as a mottled black-and-white 'garbage' image. To clear the image use the **CTRL-R** command.

Keying an asterisk in response to the background filename prompt, followed by **RETURN** , causes the option table to appear. Typing **CTRL-Q** exits the program.

Once a file has been selected and the HI-RES graphics mode is entered, single key commands are in use, and hitting **RETURN** is not necessary.

With the HI-RES screen displayed, either with the loaded background file or with a blank screen, the following set of command options becomes available. The default plotting color is white, but this may be changed as explained below.

Background Options

- I** - Moves the cursor UP
- M** - Moves the cursor DOWN
- J** - Moves the cursor LEFT
- K** - Moves the cursor RIGHT
- U** - Moves the cursor UP and LEFT
- O** - Moves the cursor UP and RIGHT

N - Moves the cursor DOWN and LEFT

, - Moves the cursor DOWN and RIGHT

P - Plot at the cursor position

T - Toggle between continuous and manual (single) plotting.

Using the **T** key to toggle the plot from single to continuous will cause a plot each time the cursor moves, regardless of the increment. When the **T** option is set to continuous, the direction key can be held down with the **REPEAT** key to provide a continuous plot.

C - Toggles Cursor ON and OFF

The cursor appears in the shape of a 'cross hair' sight with the target, at center, being the point where the dot will be plotted. The cursor is designed to change color with each single horizontal move as a guide to the color that may be expected. The actual color of the plot will not only depend on the vertical column, but also on the color of the background area.

SPACE - Locates cursor when cursor toggle is OFF

Z - Toggles cursor movement increment (1 or 8)

Sets the cursor to increment either 1 or 8 dots, depending on which increment was in use when the **Z** key is pressed.

0 - Sets the plot to BLACK

1 - Sets the plot to GREEN

2 - Sets the plot to VIOLET

3 - Sets the plot to WHITE

4 - Sets the plot to ALTERNATE BLACK

5 - Sets the plot to ORANGE

6 - Sets the plot to BLUE

7 - Sets the plot to ALTERNATE WHITE

When one of the above numeric keys is pressed, the plot color is set according to the above chart. A **0** or **4** will set the plot color to black and may be used to wipe out previous color plots. The choice of **0**

or **4** may make a difference in the 'bleed' when used next to a color. (Refer to the section on Color and HI-RES Graphics.)

X - X/Y Move

This jumps out of the plotting mode to allow repositioning the cursor to a selected value of X or Y. The present location of the cursor is shown in the TEXT portion of a mixed screen. **RETURN** (default) will leave the coordinate as is. To abort the X/Y move, type a minus one (**-1**).

L - Line Draw

Again, the program quits plotting to allow drawing a line in a selected color from one point to another. The default is to the last line constructed.

B - Block Draw

Plotting is suspended to allow making a section of the screen all one color. This may be used for large areas, then by plotting with **0** or **4**, portions may be blacked out. HI-RES has some quirks that may require a change in plans. One example is 'bleed' between colors. By reading the section on using color with HI-RES graphics, and with practice, this can be partially overcome.

S - Shape Call

This feature will call a shape from a shape table and place it at a selected position on the screen. It may be that placing a shape on top of an existing image causes strange results. For example, if a green shape is placed on a violet image the result will be white (green beside violet is white). If a green or violet shape is placed on a white (color number 3) background, the shape will not appear at all because the green and violet are already on the screen to make the white.

When **S** is used to call a shape, there are two options, either to load a shape table or get a shape from a table in memory. Any key other than a **1** or **2** returns to normal plotting mode. With the GET TABLE option, typing a non-alpha first character forces a catalog to be displayed on the TEXT screen. If a legal file name is keyed, DOS searches the disk for a binary file. If the file is not found, or it is not a binary file, the program responds with the message WRONG FILE

ask for a name again. Once a file has been successfully loaded, or if the get shape option is selected, the program checks to see that it is a 'color' table. If it is, the shape is displayed momentarily on HI-RES page two. If there is no shape table in memory, then the WRONG FILE message appears, followed by a repeat request for the file name. An acceptable response produces the prompt SCALE? which determines the scale in which the shape is to be drawn. The default scale is 1.

Once the selected shape has been viewed, the display returns to page one, with a prompt message for the X and Y coordinates at which the shape is to be placed on the background. Hitting **RETURN** accepts the X and Y of the present cursor position. To abort the placement of the shape, key a **-1** and **RETURN**. To abort and return to plotting mode in order to move the cursor to the desired shape starting position, press **X** to find the present X/Y coordinate, then **S** to again select the desired shape.

There are actually two versions of the BACKGROUND program on the SHAPES IN COLOR disk. The first is selected from the regular menu, is written in Applesoft BASIC, and supports every feature. The second, (BACKGROUND.O), is a 'compiled' version of the program which runs much faster, but which does not include the shape table loading feature. Typing **EXEC SWAP** from BASIC causes the menu to be modified so that this version is run when selected. To switch back, simply **EXEC SWAP** again. Of course, both versions can be either **RUN** or **BRUN** from the keyboard without using the menu.

CTRL-R - Wipes out the entire page to black. **Image is not recoverable.**

CTRL-Q - Exits the program but does not destroy image in memory.

F - Finished

Use **F** to end the BACKGROUND session. When the **F** key is pressed, the HI-RES display on page one is saved. (If the display has no name, the system will prompt for one.) Whether the display has been saved or not, the option of exiting or continuing the program is presented next.

Animation Using SHAPES and BACKGROUNDS

Shapes

To work with shape tables created using the SHAPES program, the shape files must be loaded into memory from a program. Once a shape table has been saved to disk, it can be called back into memory with a **BLOAD filename** command in BASIC. If the shape table needs to be loaded into a memory location other than the one in which it was stored, specify an 'A' parameter after the filename, followed by the address where it is to be loaded. Color tables created with SHAPES IN COLOR are always saved at 24600. Check the DOS manual for details on the syntax of the **BSAVE** and **BLOAD** commands.

The first 200 bytes of the color table consist of a code number and the shape color directory. The actual shape table, therefore, begins at 24800.

After a shape table has been loaded, pointers must be set to show Applesoft's shape drawing routine where the table is located. Locations 232 and 233 are the shape pointer addresses. (See page 141 of the Applesoft manual.) If a color shape file was loaded at 24600, the shape table itself is located at 24800. A single memory

SHAPES IN COLOR

21

location cannot contain a value greater than 255, therefore, the address '24800' must be broken into two parts:

```
POKE 232, 24800 - (INT(24800/256) * 256)
```

```
POKE 233, INT (24800/256)
```

If the shape table is loaded at a decimal address other than 24800, substitute that address for '24800' in the above example. All shape tables created by SHAPES IN COLOR are loaded at 24800; the decimal values to store in the pointers in this case are 224 and 96. (Performing the above arithmetic produces 244 and 96.) To set the pointers, use this code:

```
POKE 232, 224: POKE 233, 96
```

Two other statements are also important, and must be entered: rotation (**ROT=**) and scale (**SCALE=**). Unless specifically desired otherwise, **ROT** should equal 0 and **SCALE** should equal 1.

The shape color must be specified. The initial color specified using SHAPES can be found in the color directory. Otherwise, any legal value may be specified. Selecting an **HCOLOR** larger than 7, or a shape number greater than the number of shapes in the table, generates an error message.

The following programming example loads and displays shape number two of the shape file TEST:

```
10 PRINT CHR$(4); "BLOAD TEST, A24600"  
15 POKE 232,224: POKE 233,96  
20 ROT= 0: SCALE= 1  
25 HGR  
30 HCOLOR= 3  
35 DRAW 2 AT 121,100
```

Variables may be used instead, if desired:

```

5 BA = 24600:SH = 2:X = 121:Y = 100
6 REM BA IS BEGINNING OF COLOR TABLE
7 REM SH IS THE SHAPE DESIRED
8 REM X AND Y ARE THE DRAW COORDINATES
10 PRINT CHR$(4);"BLOAD TEST, A";BA
15 POKE 232,224: POKE 233,96
20 ROT= 0: SCALE= 1
25 HGR
30 HCOLOR= PEEK (BA + SH)
35 DRAW SH AT X,Y

```

If a shape color is called from the color directory as shown in line 30 above, it must be drawn on an odd-numbered-coordinate (e. g., 101 as above); otherwise, the resulting shape color will not be the same as that intended when the shape was created.

Page Switching

The Apple computer has two areas of memory that can be drawn to and plotted on using high-resolution graphics commands. These areas are called HI-RES page one and HI-RES page two. Each page is in a different area of memory, and can store a different image. Using simple POKE statements, the page to be displayed and the page to be plotted can be selected. The BASIC HGR and HGR2 commands initialize both plotting *and* display to one page *or* the other; the trick, however, to producing dramatic animation effects is to plot on one page while viewing the other.

The larger a shape, the longer it takes for the computer to draw it. Not only does this time spent in drawing the shape become apparent with large shapes, but an equal amount of time is spent 'undrawing' the shape. (That is, re-drawing it in the original background color.) This results in jerky movements.

For more flowing animation, the Apple's dual HI-RES memory areas can be used. While a shape is being displayed on page one,

prepare another behind the scenes on page two. When that picture is completed, switch the display to page two. Now the image on page one can be updated while page two is displayed. Each 'frame' of the animation is plotted on alternate pages in memory. The viewer is aware only of the instantaneous transition from one state to the next.

Adding the following code to the above example illustrates the technique of page switching:

```

40 POKE - 16302,0
45 POKE 230,64
50 CALL 62450
55 FOR I = 0 TO 2000: NEXT
300 HCOLOR= 7
305 DRAW SH AT X,Y
310 FOR I = 10 TO 100 STEP 10
315 POKE - 16299,0
320 FOR J = 0 TO 1: NEXT
325 POKE - 16300,0
330 FOR J = 0 TO 50: NEXT
335 NEXT I
340 POKE 230,32
345 DRAW SH AT X,Y
500 DEL 10,15

```

After running the above, type **TEXT** and press **RETURN** to re-enter the TEXT mode.

Movement

An example of animation *without* page switching is the GUNFIGHT program on the SHAPES IN COLOR disk. As explained above, if shapes are large, page switching creates better results. By

inserting these statements, movement is added to the previous example:

```

100 S1 = 1
105 HCOLOR= PEEK (BA + S1)
110 DRAW S1 AT 15,60
115 POKE 230,32
120 DRAW S1 AT 15,60
125 HCOLOR= PEEK (BA + SH)
130 FOR I = 0 TO 1000: NEXT
135 POKE 230,64
140 DRAW SH AT X - 2,Y
145 POKE - 16299,0
150 POKE 230,32
155 XDRAW SH AT X,Y
160 DRAW SH AT X - 4,Y
165 POKE - 16300,0
170 POKE 230,64
175 XDRAW SH AT X - 2,Y
180 X = X - 4
185 IF X < 33 THEN 250
190 POKE 230,32
195 DRAW SH AT X,Y
200 POKE - 16300,0
205 GOTO 135
250 FOR I = 0 TO 1000: NEXT

```

This code displays shape number one on the screen and moves shape number two across the screen to the left, near shape number one. Lines 300 to 345 should already be in memory from the first program.

```

400 FOR I = 0 TO 500: NEXT
405 XDRAW SH AT X,Y
410 X = 13:Y = 64
415 DRAW SH AT X,Y
420 FOR I = 0 TO 400: NEXT
425 XDRAW S1 AT 15,60
430 FOR I = 0 TO 200: NEXT
435 DRAW SH AT X,Y
440 FOR I = 0 TO 400: NEXT
445 POKE 230,64
450 CALL 62450
455 DRAW SH AT X - 1,Y
460 POKE - 16299,0
470 POKE 230,32
475 DRAW SH AT X,Y
480 FOR I = 0 TO 200: NEXT
485 POKE - 16300,0

```

It is a good practice to switch back to page one when exiting page switching routines, in the event either a **POKE -16303,0** or **TEXT** command is used to return to the TEXT mode.

With a little ingenuity, the program fragments listed above could be modified to be shorter and perhaps a little faster. However, the program was constructed in its existing form to help the programmer understand the various steps involved.

Animation

Animation not only involves moving shapes across the screen

using **DRAW** and **XDRAW**, with or without page switching, but usually consists of two or more shapes. The simplest animation involving a figure with legs, for example, would have two shapes. One shape has the legs in one position, the other shape has the legs in the other position. The move routines display first one, then the other. Another animation technique is to employ a third shape: one for the body and two other shapes for the two leg positions. With each move, the body and one of the leg positions is drawn.

For better animation, create more shapes with legs in several positions of the step cycle as illustrated in GUNFIGHT. A BASIC **FOR/NEXT** delay loop is set up which keeps the body moving smoothly. When creating the shapes, keep the body in the same column on the grid and move the leg positions, then set up the program's **FOR/NEXT** loop to increment one at a time during each step cycle.

Another method would be to keep the foot that is on the ground in the same column on the plotting grid, while the position of the body is advanced with each shape. In this program, set the **FOR/NEXT** loop to **DRAW** and **XDRAW** each shape of the animation cycle before incrementing the coordinates. GUNFIGHT uses this latter method.

Shapes with Backgrounds

If a background is loaded upon which a shape or shapes are to move about, the black in the background should be 0 for shape colors less than 4, and 4 for shape colors greater than 4. This allows the shapes to be drawn using **XDRAW**, and cleared by another **XDRAW** at the same coordinates. If the color and the black are in different groups, the xdrawn shape will not be the right color. The advantage to using **XDRAW** instead of **DRAW** to draw the shape is that the background image can be restored completely after the shape is cleared by the next **XDRAW**. If a background has both groups of colors in it (0 thru 3 and 4 thru 7), then moving a shape around the screen over the background can produce undesirable results.

Color and High Resolution Graphics

The High Resolution (HI-RES) graphics mode of the APPLE II computer can display a variety of user-designed shapes in a choice of colors. The six colors available in making high resolution graphics are: black, white, green, violet, orange and blue. Any single shape cannot contain all of the above colors. They can, however, all be displayed on the same screen. If they are all desired in a single shape, it is possible (with certain limitations) to superimpose two shapes into one. The limitations primarily involve the placement of two colors side by side.

Consider the first four colors: black (1), green (1), violet (2), and white (3), and how they are used. White is, in fact, made up of the colors green and violet placed side by side. Even-numbered horizontal plotting positions are violet, while odd-numbered positions plot as green.

For example, to produce a vertical green line, select **HCOLOR=3**. Next, plot every vertical position on an odd-numbered column. (The Apple's graphics screen is divided into 280 horizontal positions.) Plotting every other position in a row, beginning in an

odd-numbered column, produces a green horizontal line. If only the odd-numbered positions are plotted, then the entire image will be green. Plotting only the even-numbered columns produces a violet image. If two plots are placed side by side in an even and an odd column, the result is a white plot.

Conversely, any two colors which together produce white can only be plotted in their appropriate odd- or even-numbered columns, or they will not be visible. **HCOLOR=1** (green), for example, will only actually *plot* on an odd-numbered column. A plot of a vertical line on an even-numbered column simply will not show. The same is true for **HCOLOR=2** regarding odd-numbered columns.

The other group of four colors behaves in the same way as this first group. The next set of colors are black (4), referred to as alternate black, orange (5), blue (6), and white (7), referred to as alternate white. With the alternate white selected (**HCOLOR 7**), orange is plotted in the odd columns, and blue is plotted on the even-numbered columns. Plotting in adjacent columns produces alternate white, which distinguishes itself from the 'normal' white by the fact that it is composed of orange and blue, rather than violet and green.

The appearance of other shades or colors can be produced in large areas by plotting every second or fourth column in one color with every other horizontal line, then filling the horizontal lines in between with another color in every second or fourth column.

The results produced by plotting with shapes, backgrounds, plotting dots, lines, or with a block of color using a **FOR/NEXT** loop may therefore not always be as expected. The Apple's system of using two adjacently plotted colors to produce a third leaves the door wide open to distortion of the intended color.

The 'bleed' from one color into another may sometimes be cleared up by using one of the blacks, as explained below. If the programmer attempts to draw a green shape on a white (3) block, no color will show. The green is already there, in the white. **XDRAW** a green (**HCOLOR=1**) shape on a white (3) area in an odd column; the result is a violet shape. The **XDRAW** has removed the green from the white area. (Remember, **XDRAW** will redraw the shape

using the complementary color.) A green shape on a white area can be obtained by using **HCOLOR=2** (violet) then **xdrawing** it on an odd column (which erases the violet). If the shape contained green and white, what was white in the shape will now be black.

Experimenting

It will be helpful to do some experimenting with keyboard commands to see what the results are when drawing and **xdrawing** shapes.

To get a 'color' shape loaded type:

BLOAD SH6, A24600 Or any other 'color shape table.'

The pokes set the pointers to the shape table:

POKE 232, 224

POKE 233, 96

ROT = 0

Shape may be sideways if rotation is not set

SCALE = 1

Shape will be drawn to scale

HGR

HGR2 may be used, but it is full screen, and text entered at the keyboard will not be displayed.

VTAB 21

Puts cursor below graphics display

SH = 3

Sets variable SH to shape number

Attempting to draw a shape number that does not exist will generate the message, "ILLEGAL QUANTITY ERROR."

C = PEEK (24600+SH)

Looks in the color table to find the original color the shape

PRINT C

Determine what the color for this shape

If SH6 was chosen, the color is 7, alternate white.

HCOLOR = C Sets color

DRAW SH AT 51, 50 Draws an orange station wagon

Note that the x-coordinate (51) is an odd number. The shape is drawn in the color in which it was created.

XDRAW SH AT 51, 50 The shape is cleared

DRAW SH AT 50, 50 The shape is drawn in blue, because the shape was started on an even x-coordinate.

XDRAW SH AT 50, 50 The shape is cleared.

HCOLOR = 3 Change color to white (not alternate white)

DRAW SH A 50, 50 The station wagon is violet (even column)

XDRAW SH A 50, 50 The shape is cleared

DRAW SH A 51, 50 The shape is green (odd column)

XDRAW SH AT 51, 50 The shape is cleared

It can be seen by the above examples that it is possible to **DRAW** a shape on one set of coordinates, **XDRAW** it on the same coordinates, change the coordinates and then **DRAW** and **XDRAW** it again to move the shape as desired on the display.

As one shape passes over another shape on the screen, the moving shape will 'clobber' all existing shapes and ruin the display.

DRAW SH AT 51, 50

DRAW SH AT 65, 50 Note that the two are superimposed

XDRAW SH AT 65, 50 Cleared second shape, but first one is destroyed

It can be redrawn, but if a **FOR/NEXT** loop has been set up in a program to move the shape, where should a previous shape be redrawn?

DRAW SH AT 101, 50

DRAW SH AT 115, 50

XDRAW SH AT 101, 50

Note that this time shape number one was cleared, but number two was damaged.

XDRAW SH AT 51, 100 Draws green wagon

XDRAW SH AT 65, 100 Superimposes second shape on first and makes a mess of both

XDRAW SH AT 65, 50 Clears second shape and restores first

By setting up a **FOR/NEXT** loop in a program and using only **XDRAW** for shapes, the shape can usually be moved about on the screen without leaving garbage behind. However, the procedure can get a bit complicated or frustrating when working with a number of colors.

HGR Clear previous shapes

HCOLOR = C C should still be 7

DRAW SH AT 51, 50

DRAW SH AT 100, 50

XDRAW SH AT 51, 100

XDRAW SH AT 100, 100

Now there is one of each of four different colors. But using **XDRAW**, how can the **FOR/NEXT** loop discussed above work? It appears that **XDRAW** will not give either red or blue. The reason is that the black screen is the zero black. Here is where the alternate (4) black is needed.

HCOLOR = 4 Specifies alternate black

HPlot 0, 0 Sets up for full screen color

CALL 62454 Fills screen with alternate black

HCOLOR = C Color for shape

DRAW SH AT 51, 50

DRAW SH A 100, 50

XDRAW SH AT 51, 100

XDRAW SH AT 100, 100

XDRAW now works well. When working with moving shapes, experimentation is neccessary. Sometimes a change in plans is too.

Part Two Shape File Utilities

MODIFY

After selecting MODIFY, (by typing **3** at the main menu or by typing **RUN MODIFY**), the first prompt displayed is "GET WHICH SHAPE TABLE?". Typing a question mark ? catalogs the disk.

Keying RETURN advances to the option table; however, some options will not be available since there is no named file in memory. Options **4, 6** and **9**, as described below, may be used. If the filename of a shape table is entered, the program loads the shape before proceeding to the option table, with all of them available. If a name is entered that is not on the disk, or a file is selected that was not formatted as a 'color shape table' (using SHAPES IN COLOR), the system displays a message to that effect before advancing to the option table.

Transferring

A shape table is transferred from one disk to another by entering the exact name of the table to be transferred in reply to the prompt GET WHICH SHAPE TABLE? Be sure that the source disk is in the drive. As soon as the table is loaded and the option table is displayed, put the destination disk in the drive and select option **7** (SAVE). To transfer a shape table not created by the SHAPES program, see CONVERT.

Below is a list of command options that can be used in MODIFY.

Options

0 - END

Program ends and returns to the BASIC command level. To re-enter without losing the variables, type GOTO 99.

1 - SHORTEN TABLE

Shapes may be deleted from the end of the shape table, one at a time. The shape table directory will still be formatted for the maximum number of shapes originally specified, but the specified shapes at the end of the table will be gone.

2 - CHANGE NAME

The existing shape table name may be changed to any legal file name. The name must start with an alpha character and contain no commas or colons. This routine may be used to rename a shape table, then save it on another disk by using option 7.

3 - REVIEW

The review option lets you display the shapes in the table loaded in memory. The shapes are drawn in groups of twenty. Some of the shapes may overlap.

4 - GET A SHAPE TABLE

This option loads a color shape table into memory.

5 - DELETE

To delete a shape table, type **CTRL-Y**. To delete a table that is locked, use option 0 to exit the program - unlock it by keyboard command - then re-enter by **GOTO 99**, and resume.

6 - CATALOG

Displays a directory of files on the disk.

7 - SAVE

This routine saves the shape table in memory on the disk. By using option 4 with option 7, shape tables may be transferred from one disk to another.

8 - LOCK

Notice that there is no 'unlock' option. The reason is to prevent accidentally deleting a table by giving a new table the same name as an existing table.

9 - APPEND

This option is the most complex in the table as well as the most powerful. It is available even if there isn't any named program in memory; (it will ask for a name). The balance of the MODIFY section is about the append option.

Appending a Shape

To append a shape from one table to another that has room for it, enter the name of the destination table. After loading the destination table, the program prompts for the name of the source table. Enter the name of the table containing the shape you want to append onto the destination table. When the source is loaded, review it to find the desired shape number. When asked for the shape number, enter it and that shape will be displayed with the option of accepting or rejecting it. **RETURN** accepts it and appends it to the end of the destination table. Additional shapes may be appended in this way, and the new table then saved, or renamed and saved using the MODIFY option table.

Expanding a Table

Suppose a shape table was created with numbers 0 through 9 and A through Z (in upper case), and is set up for 40 shapes (36 for the characters plus a few more). There is no room left to add in punctuation or perhaps lower case.

To create a new, expanded shape table file, select a new name for the destination table. When the program looks but cannot find such a file, it prompts for the number of shapes. Enter the number of shapes desired.

The next prompt is for the source table. A question mark (?) may be entered to catalog the disk. After entering the name of the shape table to be expanded, enter the shape number to be put in the first available location of the destination table. To review the shapes in

the source table key **1** and shape number one of the source table will be displayed. A **RETURN** will accept it. Keying **RETURN** and **REPT** together will go through the source table, transfer the shape, display the option table and then repeat the procedure until all shapes have been transferred.

At this point all of the shapes from the original (source) table are now transferred to the new (destination) table. Save the new file, and add in new shapes into the expanded table using the SHAPES program.

Insert and Replace

To insert or replace shapes in an existing shape table, you need three files. The first one is the original file into which you want to insert or replace shapes. The second file is a temporary one where the shapes to be inserted are stored until they are transferred. The third file is a new file, created by combining the original with the temporary. Because shapes from the original and the temporary files are sent to this third (new) file, it is referred to as the destination file.

The name given to the destination file must be an unused one. If this file will ultimately replace the original file, you may want to give it a related name, and later delete the original file and rename the destination file. If the new file will be used separately and not replace the original shape table, give it a unique name.

To insert one or more shapes in a shape table, first put the shape or shapes to be inserted in a temporary shape table. Select the MODIFY append option, and choose a *new* name for the *destination* table. When the program fails to find the file, it will ask for the number of shapes in this new table. Enter a value large enough to accommodate both the original and the additional number of shapes. Enter the name of the table *receiving* the insertion as the *source* table. Starting with shape number one, key **RETURN** repeatedly (or with the **REPT** key) to reach the point where the shape is to be inserted. This process copies the shapes in the source (original) table into the destination (new) table. At this point, stop at the option table and select number **9** (APPEND), then key **RETURN** to accept the destination table. Key **RETURN** until

asked for the source table. Enter the name of the temporary table containing the new shapes. Select the shape number to be inserted and key **RETURN**. Then key **RETURN** again to accept each shape in the temporary table that is to be inserted at this point.

There are still more shapes to add from the original. Select option **9** (APPEND), key **RETURN**, then enter the original shape table name as the source table. After it is loaded, select the shape number in the original that follows the insertion. If the displayed shape is the right one, key **RETURN** to append that shape or key **RETURN** and **REPT** to add the balance of the original table to the destination table. Finally, select option **7** (SAVE) to save the new table on the disk.

Technical Details on Shapes in Color

Shape tables created using the SHAPES program vary slightly from the standard shape table as described on pages 92-97 of the Applesoft manual. One of the variations is that a color directory of 199 bytes is set up immediately preceding the shape directory. Because of this, the components of a shape table not created by the SHAPES program may be in the wrong location for use with SHAPES IN COLOR.

To work with a previously prepared shape table, use the CONVERT program to change it to the format required by SHAPES IN COLOR.

Another variation is that the last two bytes of the shape directory (just ahead of the first shape) in a standard shape table are used to point to the location of the last shape in the table. The SHAPES IN COLOR directory has two additional bytes that point to the location of the next shape to be created. This feature identifies the end of the existing shape table, whether or not that table is full.

When the shape tables that are saved by the SHAPES program are loaded at their normal location (24000), the first directory (the color directory) begins at decimal address 24600 and extends through 24799. The byte at 24600 contains the number 192 to identify the binary file as a color table; location 24801 contains 193 for the same purpose. The actual color directory starts at 24601 with the original color (0 - 7) of the corresponding shape.

Consider the CANDLE project in the appendix. The shape color initially used is the alternate white. When alternate white is selected as the initial color for the shape, two things happen. First, **HCOLOR** is set to 7 in the program itself for plotting the shape while it is being drawn. Next, the number 7 is poked into 24601 in the shape color directory (24600 plus the shape number, which in this case is one). Since this color directory is saved with actual shape table, any program written using shapes created by the SHAPES program can call up the original color in which the shape was designed. The following code will do this:

```
HCOLOR = PEEK(24600 +SH) : DRAW SH AT X,Y
```

SH is the shape number, X is the odd x-coordinate and Y is the y-coordinate. The SHAPES program required that the shape be started on an odd numbered column to assure its being drawn in the right color when called from memory. If the shape was drawn in either white and the resultant color is not important, then the shape may be drawn on either odd or even numbers.

The actual shape table (including the shape directory) starts at decimal address 24800. This is where the shape table pointers must point to. When the table is saved or loaded, the beginning address of the binary file is 24600. For this reason, if a shape table that was not created the SHAPES program is loaded using SHAPES IN COLOR, it is loaded a 24600, where the color directory starts. The pointers locate the address as 24800, 200 bytes too high.

When the MODIFY program is run and the Append option (9) is selected, the destination table including the color table is placed at 24600, just above page two of the HI-RES screen. The source table is placed in the area of the page two HI-RES screen and 200 bytes ahead of the actual shapes table, to include the color table. If a standard shape table was chosen for the source table, the pointers to that table will be 200 bytes above where it actually starts. However, if the MODIFY program does not find the numbers 192 and 193, the table will be loaded at an address 200 bytes higher so that the first address referred to by the pointers will be the number of shapes in the directory. As shapes are transferred from the source table to the destination table, the table will be set up in the SHAPES IN COLOR format.

CONVERT

The CONVERT program was added to the disk to make it easier for the user to manipulate, or add to, previously constructed (standard, or non-color) shape tables. When a binary file is loaded the program looks for identifying clues to determine if it is in 'color shape table' format or a 'standard' format. If it is a 'color table' format then the program will convert it to a 'standard' format and ask for the name of the file it is to be saved to. **Do not use a name that is already on the diskette that is not to be overwritten!**

The 'color table' is saved beginning with the shape directory (skipping the shape color directory), thus making the format identical to a standard table.

If the clues indicate a standard table, the program assumes it is to be converted to a 'color table,' and asks for the color of each shape as it is to be placed in the color directory; the actual shape table will be moved 200 bytes higher, and 2 extra bytes will be placed in the directory to indicate the end of the binary file.

If the clues do not indicate either a 'color table' or a 'standard table,' and it is a binary file, the program asks if it should continue.

As the 'standard' table is being transferred, each shape is displayed on the left side of the screen using the first white (color number 3) and on the right using the color that has been selected. If the shape needs a move in order to produce the desired color, a move is inserted in the shape code.

HARDCOPY

The hardcopy program was written to aid in the development of the other programs on this disk. It has been left on the disk as a convenience. It may be necessary to modify the print configuration for a given printer.

The printer slot is in line 8, and is set for slot 1. To use a printer in another slot, change **PR1#** to **PR #slotnumber**.

The program asks for the filename of the color shape table, then searches for it on the diskette. If the name entered is not a 'color table,' the program prompts again for a name. A **CTRL-Q** will exit.

When the file is found, the printer is turned on and a line is printed for the date, followed by the name of the shape table. The number of shapes in the table, followed by the length of the binary file, which includes the color table, the shape directory, and the shapes is printed next. The color table is printed to show the color number that each shape was created using. The color table starts at 24601. 24600 contains 192, and is used as part of the 'color table' identification.

Part Three

Appendix A: SHAPES Program Commands

General

T

- toggle

R

- review shapes in shapes table in groups of twenty

CTRL-Q

- exits program

GOTO 99

- re-enter program after a CTRL-Q exit

REPT

- repeats the function of any movement key being pressed

Shape Building Commands

I

- moves cursor UP

M

- moves cursor DOWN

J

- moves cursor LEFT

SHAPES IN COLOR

K

- moves cursor RIGHT

P

- plots at cursor position

Z

- view HI-RES page two

RETURN

- view HI-RES page one

?

- displays length of binary table

C

- cancel shape

E

- erase last plot

!

- clear plotting grid

F

- exit plotting mode

Plot Marker Colors

1

- green

2 - violet

3

- white

5

- orange

6

- blue

7

- alternate white

Background Commands**F**

- ends the background program and saves the HI-RES page

CTRL-Q

- exits program

CTRL-R

- clears the screen if no image in memory or wipes out any image on the screen

I

- moves cursor UP

M - moves cursor DOWN**J**

- moves cursor LEFT

K

- moves cursor RIGHT

U

- moves cursor UP and LEFT

O

- moves cursor UP and RIGHT

N

- moves cursor DOWN and LEFT

,

- moves cursor DOWN and RIGHT

Z

- toggles cursor movement increment (1 or 8)

P

- plot at cursor position

T

- toggle between continuous and single plotting

C

- toggles cursor ON and OFF

SPACE

- locates cursor when cursor toggle is off.

Plot Colors**0**

- BLACK

1

- GREEN

2

- VIOLET

3

- WHITE

4

- ALTERNATE BLACK

5

- ORANGE

6

- BLUE

7

- ALTERNATE WHITE

X

- reposition cursor at newly selected X and/or Y coordinates

-1

- aborts X command

L

- draws line between two designated points

B

- draws block of a single color

S

- gets shape from shape table and puts it at selected position on screen

EXEC SWAP

- enables selections of compiled BACKGROUND.0 version;
entering EXEC SWAP again switches to regular BACKGROUND.

MODIFY Commands

0

- ends program

- shortens table by deleting last few shapes

2

- changes name of existing shape table

3

- reviews in groups of 20 all shapes in the shape table in memory.

4

- loads a color shape table into memory

5

- deletes a shape table if CTRL-Y is also keyed.

6

- catalogue

7

- saves the shape table

8

- locks a file so that it can't be accidentally renamed or deleted.

9

- append

Applesoft BASIC HI-RES Commands

To clear & display HI-RES page one:

HGR

To set page one to full screen:

POKE -16302,0

To switch from page 1 to text:

POKE -16301

To clear & display HI-RES page two:

HGR2

To set HI-RES graphics mode without HGR:

POKE -16297,0

To display page one from text mode:

POKE -16304,0

To switch from page one to page two:

POKE -16299,0

To switch from page two to page one:

POKE -16300,0

To plot on page one:

POKE 230,32

To plot on page two:

POKE 230,64

To get full screen color:

HCOLOR (0 thru 7)**HPLOT 0,0****CALL 62424**

To clear screen to black (0):

CALL 62420

To find address of BLOADEd file:

PEEK(-21902)PEEK(-21901)*256

To find length of BLOADEd file:

PEEK(-21920)PEEK(-21919)*256**ERRATA FROM PAGE 49**

To get full screen color:

HCOLOR (0 thru 7)**HPLOT 0,0****CALL 62454**

To clear screen to black (0):

CALL 62450

To find address of BLOADEd file:

PEEK (-21902) + PEEK (-21901)*256

To find length of BLOADEd file:

PEEK (-21920) + PEEK (-21919)*256

Appendix B: The Candle Shape Project

To get familiar with how to create a shape, draw an orange candle with a white flame, on a blue candleholder. Use an 18 x 20 plotting grid.

THE STEPS:

- 1 **RUN SHAPES** (or boot disk and select No. 1)
- 2 Type **CANDLE** for the shape name and touch **RETURN**. Because CANDLE is not on the disk, the number of shapes is asked for.
- 3 Touch **RETURN** for the default value of 10.
- 4 When asked for the starting coordinates, (X?), key an **N** and **RETURN** for a New Grid.
- 5 When asked for X (New Grid), key **18** (width) and **RETURN**.
- 6 When asked for Y, key **20** (height) and **RETURN**.
- 7 When returned to the plotting grid display and the prompt X?, enter an **A** and **RETURN** for the alternate white. The color bar (upper right) shows a white T with orange and blue under the arms.
- 8 Touch **RETURN** twice to accept the default values for the X and Y (1,20) starting positions. The cursor will appear in the

lower left corner of the plotting grid, which, by the way, is in the orange column. See the top of the column.

- 9 Touch **K** - the cursor moves to the right, under the blue - the color that will be used for the base. However, move up to start the handle of the holder for now.
- 10 Touch **I, I, I** - (Three times) - The cursor is moved up with each I, for a total of three vertical cells. The cursor should now be located in the 2nd column and the 4th row from the bottom. If not, touch **C** to cancel and go to step 8 to re-do.
- 11 **P** - The first plot. A small blue dot will be seen almost directly below.
- 12 **I, P, I, P** - Moves up, plots, moves up, plots. There are now 3 vertical plot markers and 3 little blue dots below.
- 13 **I, K, K** - Moves up, right, right. The cursor is now in the next blue column.
- 14 **P, K, K, M** - A plot, two right moves, and down.
- 15 **P, M, P, M, P, M** - Plot and move down 3 times.
- 16 **J, J, P** - Move left 2 times (to a blue column) and plot. The shape so far, below the grid, should have the appearance of an O, or as near a
- 17 **K, K, P, M, P** - Continues the blue right side downward. The shape looks like a blue 'q' with a short tail.
- 18 **K, K, P, K, K, P, K, K, P** - Three plot markers on the right of the 'q.' Notice that although every other cell is plotted, the shape below the grid has a solid blue line.
- 19 **K, K, P, K, K, P, K, K, P** - Three more plots with double moves puts the last plot marker in the rightmost column of the grid. And the shape has had the horizontal line extended.
- 20 **M, J, J, J, J** - The cursor is moved down once and left four times, placing it directly under the third from the right marker, still in a blue column.
- 21 **P, J, J, P, J, J, P** - Three plot markers in the bottom row, symmetrically centered with the row just above. The blue candleholder is done. Now move to a red column.
- 22 **K, I, I** - Moved right (to a red column), and up twice to clear the candle holder. A plot in the red column in the blue portion of the shape would combine to make a white dot.

- 23 **P, I, P, I, P, I, P, I, P, I, P, I, P, I, P, I, P** - plotting and moving upward in an orange column. The last plot should be in the eighth row from the top of the grid, in the same row as the second row of small dot guide markers (third row above center).
- 24 **K, K, P** - Right, right, plot.
- 25 **P** - Up, plot (still orange).
- 26 **I, J, P** - Up, left, plot (a dot in a blue column).
- 27 **I, P, K, P** - Up, plot, right, plot (adjacent plots). Notice that the shape now has a white dot, the beginning of the flame.
- 28 **I, P, J, P** - Up, plot, left, plot (more white).
- 29 **I, P, J, P** - Up, plot, left, plot (though offset to the left, still adjacent plots, and more white).
- 30 **I, P, K, P** - Up, plot, right, plot (more white).
- 31 **I, J, P** - Move up, left, plot (places an orange dot on top).
- 32 **M, M, M, P** - Moves down 3 places and plots, adjacent to a plotted column, filling in the flame a little more with a white dot.
- 33 **M, M, K, K, M, M, M** - Moves down 2, right 2, down 3, to just below a previous orange plot.
- 34 **P, M, P, M, P, M, P, M, P, M, P, M, P, M, P, M, P** - Moves and plots downward in the orange column; to, but not into, the candle holder.
- 35 Look at the actual size shape. If there were no errors, it is finished.
- 36 **F** - The grids disappear and the finished shape is drawn near the center of the screen. A **Y** or an **N** response is now required. A **Y** puts the shape in the directory.
- 37 **Y** and **RETURN** - (The plotting mode has been exited, so the return key is back in use.) The next prompt asks if CANDLE is to be saved to disk.
- 38 **Y** - The shape is saved to disk. The program will advance to the next shape.